

OPTIMASI QUERY SQL MENGGUNAKAN TEKNIK INDEXING UNTUK MEMPERCEPAT PROSES PENGAMBILAN DATA NILAI AKADEMIK MAHASISWA FAKULTAS ILMU KOMPUTER

*SQL QUERY OPTIMIZATION USING INDEXING TECHNIQUES TO SPEED UP THE
ACADEMIC GRADE DATA RETRIEVAL PROCESS OF STUDENTS OF THE FACULTY
OF COMPUTER SCIENCE*

**Uus Firdaus^{1*}, Deni Ihsan Zulkarnaen², M Yasin Aldinar³, Bagus Raihan Ali⁴
Ryan Mulyadi⁵, Frazelia arsyanda⁶, Muhammad Izdhihar Bintang Wahyudi⁷**
¹⁻⁷Universitas Djuanda

***Email Correspondence: uus.firdaus@unida.ac.id**

Abstract

In academic database systems, speed in the data retrieval process is essential as it affects overall system performance. A common problem is the slow execution of queries as data volume increases. This study aims to optimize the execution speed of structured query language queries in the process of retrieving student academic grade data by applying indexing techniques to columns frequently used for searching and filtering data. The research method involves comparative performance testing between databases with and without indexing. The results indicate that the application of indexing significantly speeds up query response times compared to execution without indexing. Thus, indexing techniques serve as an effective solution for enhancing the efficiency and performance of academic database systems in the Faculty of Computer Science.

Keywords: Database, Indexing, Query Optimization, Academic Grades, System Performance.

Abstrak

Dalam sistem basis data akademik, kecepatan dalam proses pengambilan data menjadi hal yang penting karena mempengaruhi kinerja sistem secara keseluruhan. Permasalahan yang sering terjadi adalah lambatnya eksekusi kueri ketika volume data semakin besar. Penelitian ini bertujuan mengoptimalkan kecepatan eksekusi kueri dalam proses pengambilan data nilai akademik mahasiswa dengan menerapkan teknik indexing pada kolom yang sering digunakan dalam proses pencarian dan penyaringan data. Metode penelitian dilakukan melalui pengujian performa kueri secara komparatif antara basis data dengan indeks dan tanpa indeks. Hasil pengujian menunjukkan bahwa penerapan indexing mampu mempercepat respon kueri secara signifikan dibandingkan dengan eksekusi kueri tanpa indeks. Dengan demikian, teknik indexing merupakan solusi yang efektif untuk meningkatkan efisiensi dan performa sistem basis data akademik di Fakultas Ilmu Komputer.

Kata Kunci: Basis Data, Indexing, Optimasi Kueri, Nilai Akademik, Kinerja Sistem.

PENDAHULUAN

Dalam era digital, sistem informasi akademik memiliki peranan dalam mengelola data mahasiswa, seperti data nilai, mata kuliah, serta informasi akademik lainnya. Setiap permintaan data yang dilakukan oleh pengguna baik oleh dosen, mahasiswa, maupun admin dijalankan melalui perintah SQL (Structured Query Language). Namun, ketika jumlah data semakin besar, waktu eksekusi query juga meningkat, yang dapat menurunkan performa sistem.

Salah satu teknik yang banyak digunakan untuk meningkatkan performa query adalah indexing. Index bekerja layaknya daftar isi pada buku, yang membantu sistem menemukan data dengan cepat tanpa harus membaca seluruh tabel. Dengan penerapan index yang tepat pada kolom tertentu seperti NIM atau mata kuliah, proses pencarian dipercepat secara signifikan.

TINJAUAN PUSTAKA

Query SQL (*Structured Query Language*) adalah perintah yang digunakan untuk mengakses, mengelola, dan memanipulasi data dalam sistem basis data. Dalam sistem akademik, query ini biasanya digunakan untuk menampilkan data nilai mahasiswa berdasarkan parameter tertentu seperti NIM, mata kuliah, atau semester. *Indexing* merupakan struktur data tambahan yang disimpan dalam database untuk mempercepat proses pencarian. Tanpa adanya indeks, database harus melakukan *full table scan* atau membaca seluruh baris tabel untuk menemukan hasil yang sesuai, yang berakibat pada lambatnya eksekusi query. Indeks bekerja seperti daftar isi pada buku, membantu sistem menemukan data dengan cepat tanpa harus membaca seluruh tabel. Jenis-jenis index:

- a) Primary Index: Dibuat secara otomatis pada kolom primary key.
- b) Unique Index: Mencegah adanya nilai duplikat pada kolom tertentu.
- c) Composite Index: Terdiri dari gabungan beberapa kolom untuk query dengan lebih dari satu kondisi pencarian.
- d) Full Text Index: Digunakan untuk pencarian teks yang kompleks.

METODE

Penelitian ini menerapkan pendekatan eksperimental untuk menguji dan mengukur efektivitas teknik *indexing* dalam mengoptimalkan performa query pada sistem basis data. Objek penelitian yang digunakan adalah basis data akademik Fakultas Ilmu Komputer yang dikelola menggunakan MariaDB, yang mencakup tiga tabel utama: MAHASISWA, MATA KULIAH, dan NILAI MAHASISWA. Prosedur penelitian dilakukan melalui beberapa tahap sistematis. Tahap pertama adalah analisis sistem dan perancangan query uji yang melibatkan operasi penggabungan (*join*) antara tabel MAHASISWA, NILAI MAHASISWA, dan MATA KULIAH untuk mengambil data nilai berdasarkan parameter NIM tertentu. Tahap kedua adalah melakukan eksperimen awal dengan menjalankan query tersebut pada kondisi tabel tanpa indeks untuk mengukur waktu respon dasar (*baseline*). Tahap ketiga adalah penerapan indeks secara spesifik pada kolom yang sering digunakan dalam proses penyaringan data, yaitu kolom nim dan kode MK pada tabel NILAI MAHASISWA. Tahap terakhir adalah menjalankan kembali query uji yang sama setelah penerapan indeks untuk mencatat waktu eksekusinya, kemudian membandingkan hasilnya dengan data eksperimen awal guna menganalisis tingkat peningkatan performa sistem secara komparatif.

HASIL DAN PEMBAHASAN

Hasil pengujian menunjukkan perbedaan performa yang signifikan antara penggunaan query sebelum dan sesudah penerapan teknik *indexing*. Pada pengujian awal terhadap database akademik yang berisi 10.000 data mahasiswa, query yang dijalankan tanpa indeks memerlukan waktu eksekusi rata-rata selama 3 detik untuk menampilkan data. Hal ini disebabkan oleh proses *full table scan* yang dilakukan sistem untuk

mencocokkan parameter pencarian pada setiap baris data.

Setelah diterapkan indeks pada kolom yang sering digunakan dalam proses pencarian dan penyaringan data, yaitu kolom NIM dan KODE MK, waktu eksekusi menurun drastis menjadi 1 detik. Berdasarkan data tersebut, penerapan *indexing* mampu meningkatkan kecepatan respon query hingga sekitar 70% lebih cepat dibandingkan dengan eksekusi tanpa indeks. Peningkatan ini membuktikan bahwa indeks secara efektif membantu sistem menemukan data secara langsung tanpa harus membaca seluruh tabel.

Namun, hasil pengujian juga menemukan adanya dampak pada operasi manipulasi data lainnya. Proses *insert* atau penambahan data baru ditemukan sedikit lebih lambat setelah indeks diterapkan. Hal ini dikarenakan sistem database harus melakukan pekerjaan tambahan untuk memperbarui struktur data indeks setiap kali ada data baru yang dimasukkan ke dalam tabel. Oleh karena itu, pemilihan kolom yang akan diindeks harus dilakukan secara selektif untuk menjaga keseimbangan antara performa pembacaan dan penulisan data.

KESIMPULAN

Penerapan teknik *indexing* pada sistem basis data akademik terbukti efektif dalam meningkatkan performa query SQL, khususnya dalam proses pengambilan data nilai mahasiswa. Melalui strategi pembuatan indeks pada kolom yang tepat, waktu eksekusi query dapat berkurang secara signifikan, sehingga meningkatkan efisiensi akses data pada Fakultas Ilmu Komputer. Meskipun demikian, perancangan indeks harus dilakukan secara hati-hati dan selektif karena jumlah indeks yang terlalu banyak dapat menurunkan performa operasi tulis seperti *insert* dan *update*.

DAFTAR PUSTAKA

- Connolly, Thomas, dan Carolyn Begg. 2015. *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson.
- Elmasri, Ramez, dan Shamkant B. Navathe. 2016. *Fundamentals of Database Systems*. Person Education.
- MySQL Documentation. 2024. “MySQL Index Optimization.” Diakses dari <https://dev.mysql.com/doc/>
- Silberschatz, Abraham, Henry F. Korth, dan S. Sudarshan. 2020. *Database System Concepts*. McGraw-Hill.
- Wicaksono, A. 2023. “Implementasi Teknik Indexing untuk Optimasi Query pada Sistem Informasi Akademik.” *Jurnal Teknologi Informasi Indonesia* 12 (2): 145-153.